ELSEVIER

Available online at www.sciencedirect.com
ScienceDirect

Procedia Computer Science 1 (2012) 2033–2042

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Science, ICCS 2010

A software environment for a human-aware ambient agent supporting attention-demanding tasks

Zulfiqar A. Memon[a,b], Rogier Oorburg[c], Jan Treur[a,*], Muhammad Umair[a,d], Michael de Vos[c]

[a]*VU University Amsterdam, Department of Artificial Intelligence, De Boelelaan 1081, 1081 HV Amsterdam*
Email: {zamemon, treur, mumair}@few.vu.nl URL: http://www.few.vu.nl/~{zamemon, treur, mumair}
[b]*Sukkur Institute of Business Administration (Sukkur IBA), Airport Road Sukkur, Sindh, Pakistan*
[c]*Force Vision Lab, Barbara Strozzilaan 362a, 1083 HN Amsterdam, The Netherlands Email:{ rogier, michael}@forcevisionlab.nl*
[d]*COMSATS Institute of Information Technology, Department of Computer Science, Lahore, Pakistan*

Abstract

This paper presents a software environment for a human-aware ambient agent providing support for a human performing an attention-demanding task. The agent obtains human attention-awareness by use of a dynamical model of human attention, gaze sensoring by an eye-tracker, and information about features of the objects in the environment. It has been implemented in a component-based, event-driven manner within the Adobe® Flex® environment, thereby also integrating the Tobii® eye-tracker. It has been applied in a setup for a task where the human has to identify enemies and allies, and eliminate the enemies.

Keywords: Human-aware; ambient; agent; attention; Adobe Flex

1. Introduction

One of the more ambitious challenges for the design of ambient or pervasive agent systems is to create an appropriate representation and awareness of a human's states; e.g., [1, 2, 3, 4]. Such human-aware agent systems can be taken to perform a certain type of mindreading or to possess what in the psychological and philosophical literature is called a Theory of Mind; e.g., [5, 6, 7]. Like in the evolutionary human history, within different applications such mindreading may address different types of human states, such as intention, attention, belief or emotion states; e.g., see [6]. The application focus of the work reported here is on attention-demanding tasks. Some literature on attention models can be found, for example, in [8, 9, 10, 11]. The ambient software agent has been designed in such a way that it has awareness of the human's attention states in particular. This awareness is built up using three ingredients: (1) information obtained by sensoring of the human's gaze by an eye-tracker, (2) information on attention attracting features of objects in the environment, and (3) a differential equations model of the dynamics of attention levels over time integrating the instantaneous information from (1) and (2), and using

* Corresponding author. Tel.: +31-20-598-7763; fax: +31-20-598-7653.
E-mail address: treur@few.vu.nl.

doi:10.1016/j.procs.2010.04.227

partial persistency of attention over time.

Within the ambient agent the estimated human attention levels for different objects are compared to estimated levels of urgency of the objects present, which are interpreted as the attention demands. Based on this comparison between attention supply and attention demand a set of objects that need intervention is determined, and for these objects appropriate intervention actions are selected.

The agent-based ambient support software environment has been implemented within the Adobe® Flex® development environment according to a component-based design, with event-driven interaction between components based on ActionScript. For the gaze sensoring it integrates the Tobii® eye-tracker. Before entering a task performance session, initially it allows to set values for a number of parameters.

In this paper, first an example of the type of application is presented. Next the assessment of the attention and how the intervention actions are generated is discussed. Later, some more details of the developed software are discussed. The paper is concluded with a discussion.

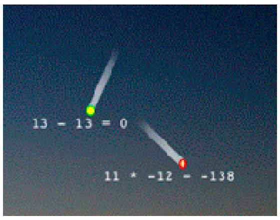
2. The Type of Attention-Demanding Task Considered

To address the possibilities for a human-aware ambient agent supporting attention-demanding tasks a specific environment has been developed. Main characteristics of the type of task considered in this environment are:

- it has elements of monitoring, inspection and analysis of a visually represented environment
- it has elements of deciding on and performing actions to cope with circumstances that require intervention
- part of the task concerns cognitive load
- (visual) attention plays an important role
- work load may vary over time and at times may become stressful and exhausting

This context has been given the form of a (simplified) simulated environment to perform such a type of task. More specifically, the idea is that a person has to (1) inspect visually displayed moving objects in the environment, and identify whether such an object is dangerous (enemy) or not (ally), and (2) for each of such objects, depending on the identification perform some actions. The person in charge (further called player) faces objects traversing the screen from the top to the bottom.

The player’s task consists in classifying each object as ally or enemy, and shooting down the enemies while letting allies land safely. Identification of an object is a cognitive task, in a simplified form represented by an arithmetical calculation. In order to determine whether an object is an ally or an enemy, the player mouse-clicks on the object where after an arithmetical formula appears next to that object. A correct formula means that we are dealing with an ally while enemies show false formulas. To categorize an object as ally, the ← key can be pressed, for enemies the → key. A spoken voice will confirm the choice. Objects designated as allies turn greenish yellow while those identified as enemies go reddish orange. The player uses cannon at the bottom of the screen to shoot down (hostile) objects. Fig 1 shows an identified ally to the left and an identified enemy to the right. Using scores and costs that are assigned it can be investigated if support measures affect the efficiency of the task execution. Scores are assigned according to the table shown in Fig. 1. Costs are counted as +1 per fired missile; costs are calculated and shown independently of the score.



Objects	Ally	Enemy
landed	+1	-1
shot down	-1	+1

Fig 1. Part of the screen and score table

3. Assessment of the Human's Attentional State

The assessment of the human's attentional state concerns three different elements. First, from the supply side it is discussed how it is estimated much attention the human is paying to each of the different objects. Next, from the demand side it is discussed how it is estimated how much attention is required for each of the objects (urgency). Finally, the two types of estimations are compared to determine a discrepancy assessment for each of the objects.

3.1. Attention Estimation

To perform a demanding task of the type considered, it is important to have sufficient visual attention. In such a situation, a player may be assisted by an ambient agent that keeps track of where his or her attention is, and provide support in case the attention is not where it should be. To this end, the software exploits a domain model of the state of attention of the player over time. Below the domain model for attentional processes of the player is described, as adopted from [8].

The player's attention at the current point in time is distributed over all objects on the screen (incoming unidentified objects, the object that displayed formulas, identified ally- and enemy-indications). Some objects get more attention, while other objects get less attention. This depends on two main aspects. One is the distance between the object and the player's gaze direction: the shorter the distance between the gaze target and the object, the higher attention level of this object is, and vice versa. Another main aspect is an object's potential (capability) for attracting attention based on the object's characteristics (such as brightness and size, for example: a brighter and larger object attracts more attention). As a first step for each object on the screen its potential for attracting attention is determined. For example, when object O has two characteristics, brightness and size, which have numerical values V_1 and V_2, respectively, then the value of the potential of the object O for attracting a player's attention at a given time point is calculated as $w_1 * V_1 + w_2 * V_2$ where w_1 and w_2 are parameters (weights) reflecting the relative importance of the characteristic and size, respectively; w_1 and w_2 are both real numbers between 0 and 1. When multiple characteristics C_i for $i = 1, \ldots, m$ are involved, the attention-attracting potential of an object O is expressed by a weighted sum

$$\sum_{i=1}^{m} w_i * V_i$$

where V_i is the value of characteristic C_i. The precise choice of the characteristics together with the values of the corresponding weights (the sum of which is set on 1 by normalisation) can be chosen from the following list: size, brightness, colour, blinking of the object, a circle around an object, blinking of circle around object, time to the ground, direction of movement (angle), speed, known as enemy or ally, distance to the ground. The values for these variables are expressed by numbers in the interval $[0, 1]$, with 0 no attraction potential and 1 highest attraction potential. For a specific session, initially a selection of them can be made and weight factors assigned.

As a next step the player's gaze location is taken into account, in order to combine it with the potential of an object for attracting the player's attention; a location is indicated by a point on the screen, represented by a pair of coordinates (x,y) with respect to horizontal and vertical axes for the screen. To take gaze into account the potential of the object for attraction player's attention is divided by a function depending on the player's gaze location:

$$V(O) = \left(\sum_{i=1}^{m} w_i * V_i \right) / (1 + \alpha * d(O, G)^2)$$

Here $d(O, G)$ is the Euclidean distance between the object O and the player's gaze G. If the gaze is represented by coordinates (x_1, y_1) and the object's location coordinates are (x_2, y_2) then $d(O, G)$ can be determined as $\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$ and hence $d(O, G)^2$ by $(x_1-x_2)^2+(y_1-y_2)^2$. The parameter α (represented by a positive real number) affects how fast an object loses the player's attention as the gaze moves further away from the object. The higher α is, the lower the attention for objects distant from the player's focus.

It is assumed that a person can have a fixed total amount of attention A distributed over all available objects on the screen. Therefore the attention level $AV(O)$ of an object O expresses the amount of the player's attention directed at the object O as a proportion (percentage) of the player's total attention. If there are n objects in total on the screen and the attention value $V(O_j)$ of the object O_j is indicated by V_j $(1 \leq j \leq n)$, then the attention level $AV(O_i)$ of the object O_i is determined in a normalised form as

$$AV(O_i) = (V_i / (V_1+V_2+\ldots+V_n)) * A$$

As a gaze can change from one moment to the other, and it has a strong effect on the attention values, these attention values can show quite instable patterns. However, it can be assumed that attention persists over shorter time periods. To model this the attention values for object O can be modelled with persistence as $PAV(O)$ using the following difference equation:

$$PAV(O)_{t+\Delta t} = PAV(O)_t + \beta*(AV(O)_t - PAV(O)_t)\Delta t$$

Here β is an attention flexibility parameter with a positive value between 0 and 1, and time step with $0 \leq \Delta t \leq 1$; a high value of β results in fast changes and a low value in a high persistence of the old value. Note that for $\Delta t = 1$ this can be rewritten into a weighted sum form

$$PAV(O)_{t+1} = \beta*AV(O)_t + (1-\beta)\, PAV(O)_t$$

which shows more explicitly how the parameter β expresses partial persistence of attention; for example, for the extreme value $\beta = 0$, the attention state would be fully persistent. Written in differential equation format the dynamical model is as follows:

$$dPAV(O)/dt = \beta*(AV(O) - PAV(O))$$

This represents a system of n differential equations for all of the n objects involved, which via $AV(O)$ integrates the gaze information and information about the object features over time.

3.2. Urgency Estimation

The task considered involves high attention demands, especially when many objects are coming in a short period of time. Some of these objects can be ignored, for example because they are allies. Other objects demand more attention, for example enemies that have a high speed and/or that are already close to the ground. To find out which objects should be given attention, it is estimated how critical objects are: the urgency of objects. This is done based on n urgency-indication factors by a weighted sum.

$$UV(O) = \begin{cases} \sum_{i=1}^{n} w_i * U_i & \text{if } O \text{ is an enemy} \\ 0 & \text{if } O \text{ is an ally} \end{cases}$$

Here U_i is the value of the i-th urgency factor and w_i the weight of this factor (with total sum 1). The factors that can be considered here are: time to the ground, distance to the ground, direction of movement (angle), speed. The values for these variables are expressed by numbers in the interval $[0, 1]$, with 0 no urgency and 1 highest urgency. For a specific session, initially a selection of them can be made and weight factors assigned. It has been assumed that like attention also urgency persists over shorter time periods. To model this the attention values for object O can be modelled with persistence as $PUV(O)$ using the following difference equation:

$$PUV(O)_{t+\Delta t} = PUV(O)_t + \gamma*(UV(O)_t - PUV(O)_t)\Delta t$$

Here γ is an urgency flexibility parameter with a positive value between 0 and 1, and time step with $0 \leq \Delta t \leq 1$; a high value of γ results in fast changes and a low value in a high persistence of the old value of the urgency. Note that for $\Delta t = 1$ this can be rewritten into a weighted sum form

$$PUV(O)_{t+1} = \gamma*UV(O)_t + (1-\gamma)\, PUV(O)_t$$

which shows more explicitly how the parameter γ expresses partial persistence of urgency; for example, for the extreme value $\gamma = 0$, the urgency state would be fully persistent. Written in differential equation format the dynamical model is as follows:

$$dPUV(O)/dt = \gamma*(UV(O) - PUV(O))$$

This represents a system of n differential equations for all of the n objects involved, which via $UV(O)$ integrates the gaze information and information about the object features over time.

3.3. Discrepancy Assessment

To determine whether there is enough attention for objects that demand attention some comparison has to be made. The attention levels that are estimated can be considered as offered attention utilisation, and the urgencies can be considered as demands. However, in principle these quantities are not expressed according to a measure such that they are comparable. For example, the total sum of attention values for all objects is A, which may be set on 1, whereas the total sum of urgencies can easily be much higher than 1. Moreover, in general, it is not clear at forehand how high an attention level has to be in order to represent ‘enough attention’. Therefore some rescaling has been made in the comparison, in the following discrepancy assessment:

$$D(O) = w_u * U(O) - w_a * PAV(O)$$

This uses initially set weight factors for urgency and attention level to deternine the discrepancy for an object O. The interpretation is that $D(O) = 0$ means sufficient attention for the object, $D(O) < 0$ more than sufficient, and $D(O) > 0$ insufficient attention. These discrepancy assessments are used to determine appropriate intervention actions.

4. Selection Process for Intervention Actions

Within the process to determine intervention actions two main decisions made are based on the assessment information: (1) which objects to address, and (2) which intervention actions to apply on the addressed objects. Below the criteria are discussed on which these decisions are based. For a summary, see Table 1.

4.1. Object focus set

For the first decision two criteria are considered, based on two parameters th (a real number ≥ 0) and k (a natural number): (a) the discrepancy of the object is $> th$, and (b) it is among the k objects with highest discrepancy. Given values set for these parameters, the combination of these criteria defines a focus set of objects addressed. A special case used in the example simulations shown is $k = 1$ and $th = 0$. In this case the object with highest positive discrepancy is selected.

4.2. Intervention action selection

Action selection does not involve only (a) the selection of a specific type of action (for example, to display a pointer to the object), but also (b) determination of the intensity value for that action (e.g., the size of the pointer). The actions that may be considered are affecting the following variables: brightness, colour, size, blinking, a circle around an object, blinking of circle around an object.

Table 1. Decision criteria for selection of intervention action

	Indication	Decision parameters	Decision criteria
Object focus set	objects with high discrepancy	discrepancy D threshold th number of objects k	$D > th$ & the object is among the k objects with highest D
Selected action	actions with high impact on discrepancy	current intensity value X sensitivity S	action with highest maximal impact $- S*(1-X)$
Action intensity	approximate compensation of discrepancy	discrepancy D current intensity value X sensitivity S	action intensity $min(1, X - D / S)$

The values for these variables are expressed by numbers in the interval [0, 1], with 0 no intensity and 1 highest intensity. For each object in the focus set an action and its intensity value is selected using *sensitivity factors*. A sensitivity factor S for a variable X addressed by a certain action with respect to discrepancy D defines how much (in the given situation) the discrepancy D will change upon a change in the variable, which is mathematically denoted by the partial derivative $\partial D/\partial X$. To determine a sensitivity factor S (i.e., determining the partial derivative $\partial D/\partial X$) both analytical and approximation methods or a combination of them can be used. The attention model is defined by a differential equation and no direct formula is given as, for example, was the case in the adaptation approach described in [12]. Therefore here this partial derivative cannot be determined by analytic calculation. As an approximation method, a small arbitrary change ΔX in the value of variable X can be tried (for example a change of 4%, which for $X = 0.5$ makes $\Delta X = 0.02$), and based on the resulting change ΔD in the value of D (for example $\Delta D = -0.15$) found in predicted discrepancy, the sensitivity factor S can be estimated by $S = \Delta D / \Delta X$ which for example provides $S = -0.15 / 0.02 = -7.5$). Note that the norm for the discrepancy D is 0, so $\Delta D = D$ can be taken.

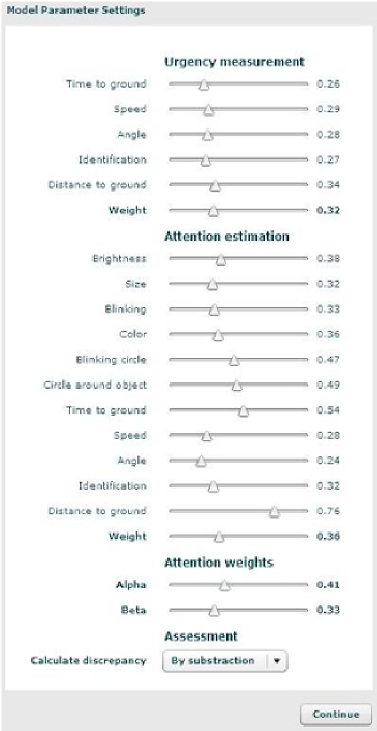


Fig 2. Interface to set parameter values

Given that for each action option the sensitivity factor is known, as a negative real number, it is used in the following manner. First it is multiplied by the remaining interval within [0, 1] for the variable addressed by the action: this provides the maximal effect $-S*(1-X)$ on D that can be achieved by applying this action. An action is chosen where this value is maximal, while its potential impact on discrepancy is the highest. To approximately compensate the discrepancy the following value is taken for the intensity of the action: $\Delta X = -D/S$, and $X + \Delta X = X - D/S$. So, when $D = 0.6$, $S = -3$ and X has value 0.3 this obtains $\Delta X = 0.6/3 = 0.2$, so the action is selected with value $X + \Delta X = 0.5$. In case $X - D/S$ exceeds 1, the maximal *intensity 1 can* be taken for the action. Table 1 shows an overview of the decisions made in intervention action selection.

Table 2. Values of parameters and variables for an example session

	Urgency weight w_1	Urgency value U(O)	Attention weight w_2	Attention value PAV(O)	Discrepancy D(O)
object1(leftmost)	0.32	0.8	0.36	0.2	0.184
object2(middle)	0.32	0.6	0.36	0.8	-0.096
object3(rightmost)	0.32	0.4	0.36	0.3	0.02

5. The Software Environment in Adobe Flex

In this section some details of the software environment for ambient support of attention-demanding tasks are described. It has been implemented within the Adobe® Flex® development environment, according to a component-based design. Between the different components event-driven interaction takes place, implemented using

Fig 3. Snapshot of an example session

ActionScript. Moreover, for the gaze sensoring a specific sensoring component was included that connects the Tobii® eye-tracker used. An interface component allows to initially set parameter values. Two other components represent the assessment and action selection processes described in Sections 3 and 4. A number of sessions have been performed using this software environment to test the behaviour, and some data of one example session have been included below. Fig 2, shows the interface component where the user sets the values of the parameters used in the Attention estimation, Urgency estimation and Discrepancy assessment as explained in Section 3. Table 2 shows the values of urgency and attention estimation, and discrepancy of three objects during the session. Example results of the session are shown in the snapshot displayed in Fig 3, where 3 objects are visible; i.e. object1 (leftmost object), object2 (middle object) and object3 (rightmost object) with different types of support provided to them depending upon the estimated urgency and attention levels. The gaze as given by the Tobii® eye-tracker is denoted by "╬". As can be seen from the data shown in Table 2, object2 has negative discrepancy value (i.e., sufficient attention is given), so in line with the decision criteria described in Table 1 in Section 4, no support is provided to it. In this situation support is provided to the object having highest discrepancy, i.e., object1. Fig 3 also shows the current score and the cost in this stage of the session. As can be seen, because of the support measures provided the cost is less as compared to the score earned by the human, which shows efficiency of the task execution with the given support.

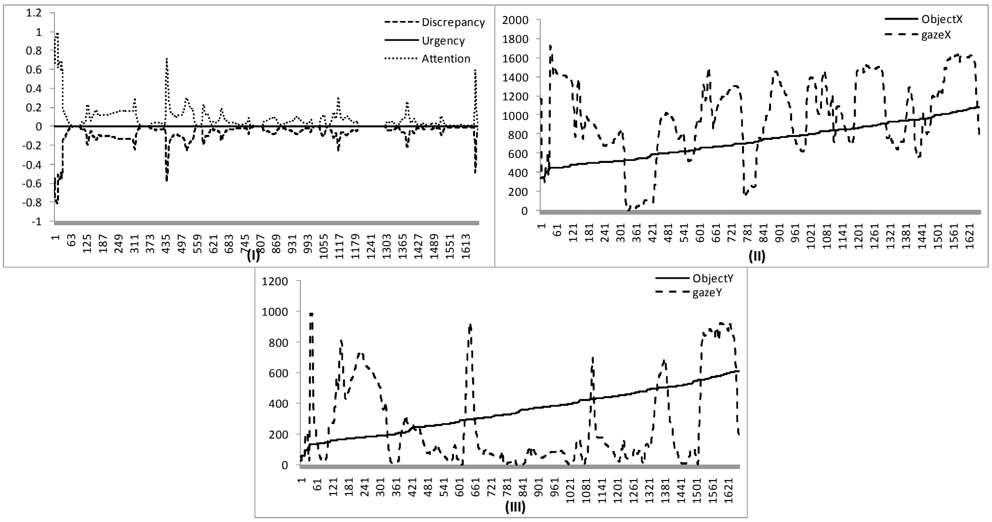


Fig 4. Simulation trace 1 - for an ally object (α=0.9, β=0.95, γ=0.1, w_1= w_2=0.95)

6. Simulation Experiments

Based on the ambient support system described above, a number of simulations have been performed. A first example simulation trace for a non-attentive person included in this section is shown in Fig 4 as an illustration (one ally object) and Fig 5 (one enemy object). In all of the figures shown, time is on the horizontal axis, whereas in Fig 4 (I) and Fig 5 (I), the vertical axis shows the values of the human's attentional state elements, and in Fig 4 (II) & (III) and Fig 5 (II) & (III) it shows screen coordinates. The example trace in Fig 4 (I) shows the values of the elements of human's attentional state for an ally object. As can be noticed the urgency value of an ally object remains 0 for the whole simulation session, as has been described in the section of urgency estimation. It further shows that urgency value is independent of human's gaze. As described in the section on Attention Estimation, the attention element depends on the difference between the human's gaze value and current location of the object. This can be seen in Fig 4 (I) and (II) & (III), where the attention value increases as the gaze of the human comes closer to the object and it decreases as human gaze goes away from it. Fig 4 (I) also shows the discrepancy element, which is the weighted difference between the attention and urgency elements.

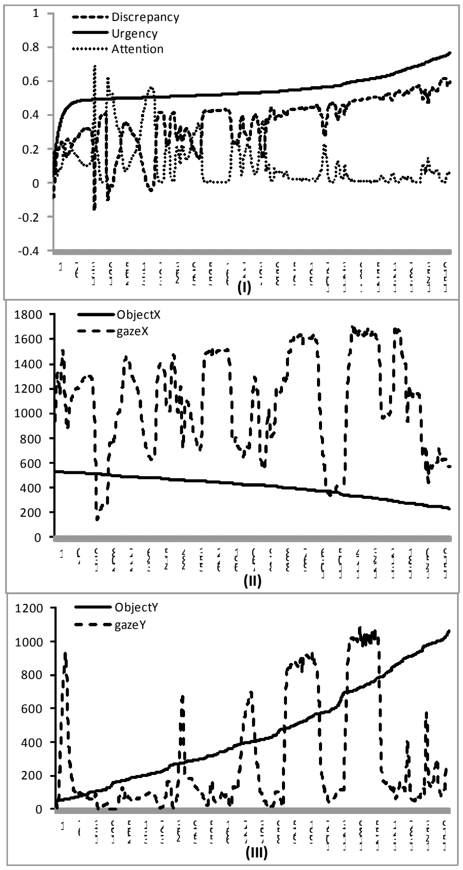


Fig 5. Simulation trace 1 – for an enemy object
(non-attentive person; α=0.9, β=0.95, γ=0.1, w_1= w_2=0.95)

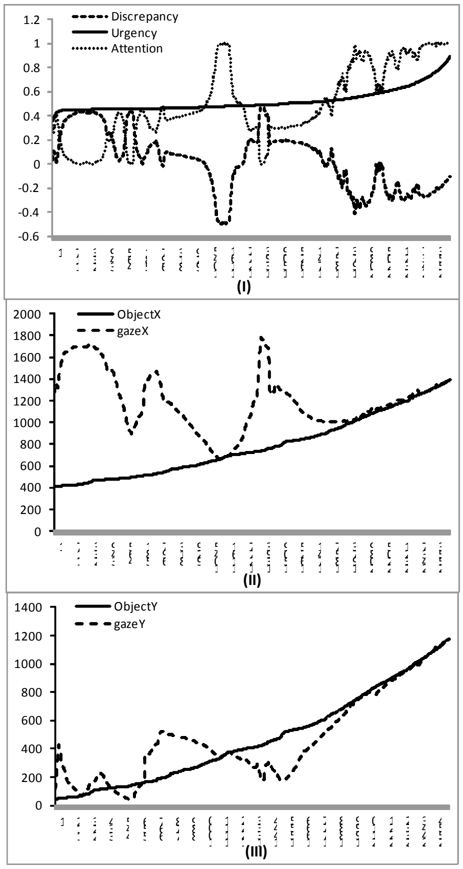


Fig 6. Simulation trace 2 – for an enemy object
(attentive person; α=0.9, β=0.95, γ=0.1, w_1= w_2=0.95)

For this particular example simulation the weights for attention and urgency are same, i.e., 0.95, and as the urgency for this particular object is 0 (because of being an ally), therefore the discrepancy value is exactly reciprocal to the attention value. In other words, for this simulation experiment the discrepancy is dependent on attention value only. The example trace in Fig 5 (I) shows the values of the elements of human's attentional state for an enemy object. As can be noticed the urgency value of an enemy object gradually increases over the time until it reaches to ground, as has been described in the section of urgency estimation. For this example simulation the factors that have been considered are time to ground and distance to ground along with object identification. It can also be noted that as the object's Y coordinate increases, i.e., the object is coming closer to ground (see Fig 5 (III)), the urgency value increases simultaneously On the other hand, the value of the attention element shows the similar pattern as for an ally object described earlier, because it depends on the difference between the gaze and object coordinates. This can be seen in Fig 5 (I) and (II) & (III).

This example simulation trace depicts a scenario where the human does not paying attention to the object even though the urgency value of the object increases. This can be seen in Fig 5 (II) & (III) where the user gaze goes away from the object. Fig 6, shows the simulation of an example scenario where the person pays attention to those objects for which the proposed software environment provides support, i.e., those objects that are closer to ground and has not been paid attention to. This can be seen in Fig 6 (I), (II) & (III), where the person's gaze comes closer to the object whose urgency is higher because of the ambient support provided by the software, as opposed to the pattern shows in Fig 5.

The simulation experiment trace shown in Fig 7, compares two enemy objects, where the person pays attention to the most urgent object as compared to the least urgent one. As can be noticed in Fig 7 (I), the person initially pays attention to the obj1 but loses attention after a short duration afterwards. Later, as the proposed software environment provides support to the most urgent object, i.e., obj1, the person again starts to pay attention to it, and hence the attention for that object increases for the rest of the simulation trace. Notice that the attention value of obj2 is lower as compared to obj1. It can also be validated by the generated pattern of movement of the human gaze and the object coordinates, as can be seen in Fig 7 (II) & (III).

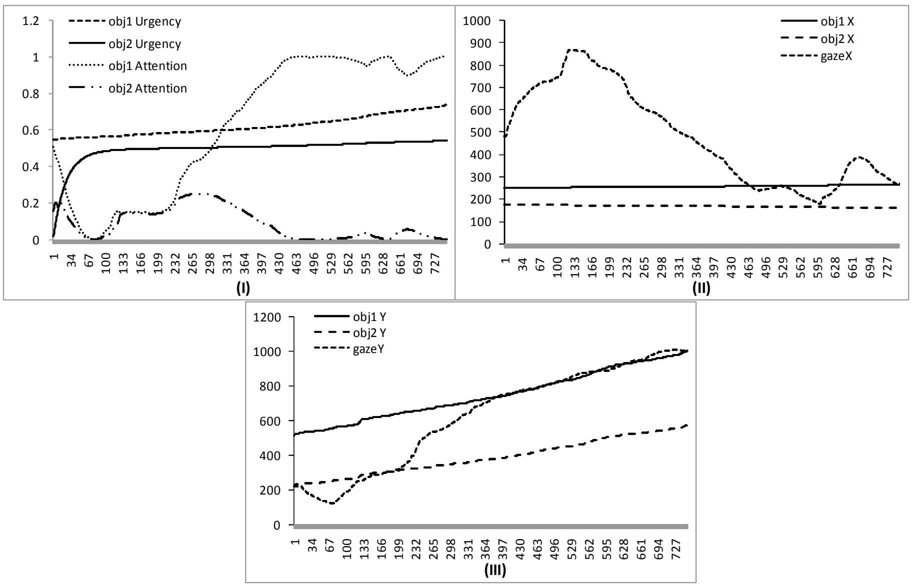


Fig 7. Simulation trace 3 – Comparison of two enemy objects (α=0.9, β=0.95, γ=0.1, w_1= w_2=0.95)

7. Discussion

To function in a knowledgeable manner, ambient or pervasive agents (e.g., [1, 2, 4]) need to perform some form of mindreading (e.g., [5, 6, 7]) to obtain a model of the human(s) they are supporting. The agent-based software environment presented here focusses on mindreading concerning a human's attention states (e.g., [8, 9, 10, 11]), based on information acquired by sensoring of the gaze, features of the relevant objects in the environment, and a dynamical model based on differential equations integrating all this information. For the attention estimation the software agent adopts the model described in [8], which was also used in [13] and [14]. In contrast to [13] and [14], in the approach presented here the selection of intervention actions (as summarised in Table 1) is based on numerical approximation methods using sensitivity factors; these numerical methods are different from and more generally applicable than the analytical approach used in [13] and [14], where within the action generation process no differential equation format for (partial) persistency of attention levels is incorporated and no dynamic comparison between action options is made. The software environment was implemented according to a component-based design within the Adobe® Flex® development environment, which makes it easy to adapt. Interaction between components was implemented using ActionScript, in an event-driven manner. A sensoring component was included for the gaze sensoring which connects to the Tobii® eye-tracker.

The developed agent-based system was evaluated through a number of experiments, some of which were discussed in the paper. It was shown that after intervention actions, attention for urgent objects was higher. Future work will address the combination of the model for attention with a model that estimates the human's work pressure and exhaustion and its effect on the attention.

References

1. Aarts, E.; Collier, R.; Loenen, E. van; Ruyter, B. de (eds.) (2003). Ambient Intelligence. Proc. of the First European Symposium, EUSAI 2003. Lecture Notes in Computer Science, vol. 2875. Springer Verlag, 2003, pp. 432.

2. Aarts, E., Harwig, R., and Schuurmans, M. (2001). Ambient Intelligence. In: P. Denning (ed.), The Invisible Future, McGraw Hill, New York, pp. 235-250.

3. Bosse, T., Hoogendoorn, M., Klein, M., and Treur, J., A Component-Based Ambient Agent Model for Assessment of Driving Behaviour. In: F.E. Sandnes et al. (eds), Proc. of the 5th International Conference on Ubiquitous Intelligence and Computing, UIC'08. Lecture Notes in Computer Science, vol. 5061. Springer Verlag, 2008, pp. 229-243.

4. Riva, G., F. Vatalaro, F. Davide, M. Alcañiz (eds.) (2005). Ambient Intelligence. IOS Press.

5. Bosse, T., Memon, Z.A., and Treur, J., A Two-Level BDI-Agent Model for Theory of Mind and its Use in Social Manipulation. In: Proceedings of the AISB 2007 Workshop on Mindful Environments, 2007, pp 335-342.

6. Gärdenfors, P. (2003). How Homo Became Sapiens: On The Evolution Of Thinking. Oxford University Press, 2003.

7. Goldman, A.I. (2006). Simulating Minds: the Philosophy, Psychology and Neuroscience of Mindreading. Oxford University Press.

8. Bosse, T., Maanen, P.-P. van, Treur, J., Simulation and Formal Analysis of Visual Attention, Web Intelligence and Agent Systems Journal, vol. 7, 2009, pp. 89-105.

9. Itti, L. and Koch, C., Computational Modeling of Visual Attention, Nature Reviews Neuroscience, vol. 2, no. 3, 2001, pp. 194-203.

10. Parkurst, D., Law, K., and Niebur, E. Modeling the role of salience in the allocation of overt visual attention. Vision Research Journal, vol. 42, 2002, 107-123.

11. Turatto, M., and Galfano, G., Color, form and luminance capture attention in visual search. Vision Research, 40, 2000, 1639-1643.

12. Bosse, T., Memon, Z.A., and Treur, J., Adaptive Estimation of Emotion Generation for an Ambient Agent Model. In: Aarts, E., Crowley, J.L., Ruyter, B. de, Gerhäuser, H., Pflaum, A., Schmidt, J., Wichert, R. (eds.), Ambient Intelligence, Proceedings of the Second European Conference on Ambient Intelligence, AmI'08. Lecture Notes in Computer Science, vol. 5355. Springer Verlag, 2008, pp. 141–156.

13. Bosse, T., Lambalgen, R. van, Maanen, P.P. van, and Treur, J., Attention Manipulation for Naval Tactical Picture Compilation. In: Baeza-Yates, R., Lang, J., Mitra, S., Parsons, S., Pasi, G. (eds.), Proceedings of the 9th IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT'09. IEEE Computer Society Press, 2009, pp. 450-457.

14. Bosse, T., Lambalgen, R. van, Maanen, P.-P. van, Treur, J., Automated Visual Attention Manipulation. In: L. Paletta, Tsotsos, J.K. (eds.), Attention in Cognitive Systems, Proc. of the Fifth International Workshop on Attention in Cognitive Systems, WAPCV'08. Lecture Notes in Artificial Intelligence, vol. 5395. Springer Verlag, 2009, pp. 257-272.